PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	CHAMBLISS et al.	Examiner:	ANYA, C..
Serial No.:	10/603,327	Group Art Unit:	2194
Filed:	June 25, 2003	Docket No.:	**SJO920030023US1** (IBMS.005-0524)
Title:	METHOD, APPARATUS AND PROGRAM STORAGE DEVICE FOR PROVIDING A TWO-STEP COMMUNICATION SCHEME		

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence and the papers, as described hereinabove, are being deposited in the United States Postal Service, as first class mail, in an envelope addressed to: Mail Stop Appeal, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on September 7, 2007.

By: David W. Lynch

APPEAL BRIEF

Mail Stop Appeal
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This is an Appeal Brief submitted pursuant to 37 C.F.R. § 41.37 for the above-referenced patent application. Please charge Deposit Account No. 09-0466 (**SJO920030023US1**) in the amount of $500.00 for this brief in support of appeal as indicated in 37 C.F.R. § 41.20(b)(2).

09/18/2007 FHETEKI1 00000035 090466 10603327
01 FC:1402 500.00 DA

I. Real Party In Interest

The real party in interest is International Business Machines Corporation, having a place of business at New Orchard Road, Armonk, New York 10504. This application is assigned to International Business Machines Corporation.

II. Related Appeals And Interferences

Appellant is unaware of any related appeals, interferences or judicial proceedings.

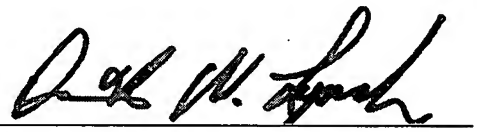
09/17/2007 FHETEKI1 00000048 090466 10603327
01 FC:1401 500.00 DA

C. Conclusion

In view of the above, Appellant submits that the rejections are improper, the claimed invention is patentable, and that the rejections of claims 1-40 should be reversed. Appellant respectfully requests reversal of the rejections as applied to the appealed claims and allowance of the entire application.

Respectfully submitted,

Chambliss, Bahner and Stophel
1000 Tallan Building
Two Union Square
Chattanooga, TN 37402
423-757-0264

By: 

Name: David W. Lynch
Reg. No.: 36,204

III. Status Of Claims

Claims 31-54 are rejected. Claims 31-54 are presented for appeal and may be found in the attached Appendix of Appealed Claims in their present form.

IV. Status Of Amendments

No amendments to the claims were made subsequent to the final rejection of Appellants' application.

V. Summary Of Claimed Subject Matter

Appellant's invention is a method, apparatus and program storage device for providing automatic performance optimization of virtualized storage allocation within a network of storage elements.

Independent claim 31 recites a program storage device (168, page 16, line 1) readable by a computer, wherein the program storage device tangibly embodying one or more programs of instructions (190, page 16, line 2) executable by the computer to perform a method for providing a two-step communication scheme. The method includes establishing a portion of memory configured to provide asynchronous, connectionless inter-process communication (240, 242, 244, page 12, lines 2-4) between a first process (e.g., 216, page 11, lines 7-8) and a second processes (e.g., 210, page 11, lines 18-19), granting exclusive access (530, page 13, line 12) to a first process to the portion of memory configured to provide asynchronous, connectionless inter-process communication (240, 242, 244, page 12, lines 2-4) between the first process (e.g., 216, page 11, lines 7-8) and the second processes (e.g., 210, page 11, lines 18-19), while having been granted to the exclusive access to the portion of memory configured to provide asynchronous, connectionless inter-process communication, accessing the portion of memory (532, page 13, line 13) configured to provide asynchronous, connectionless inter-process communication by the first process to modify the contents thereof to provide a message for processing by the second process and releasing exclusive access (536, page 13, lines 14-15) by the first process to the portion of memory configured to provide asynchronous, connectionless inter-process communication to prevent inter-process communication between the first and second process from becoming a

performance bottleneck by releasing resources of the first process after the first process modifies the contents of the portion of memory.

Independent claim 37 recites a server. The server includes a memory (192, page 16, line 3), wherein a portion of the memory (240, 242, 244, page 12, lines 2-4) is configured to provide two-step, asynchronous, connectionless inter-process communication between a first process (e.g., 216, page 11, lines 7-8) and a second processes (e.g., 210, page 11, lines 18-19), the portion of memory being configured as memory accessible by the first and second processes, wherein access to the portion of memory being granted exclusively to the first process (530, page 13, line 12) for modification of contents of the portion of memory (532, page 13, lines 13) to prevent inter-process communication between the first and second process from becoming a performance bottleneck by releasing resources of the first process after the first process modifies the contents of the portion of memory (536, page 13, lines 14-15).

Independent claim 43 recites a system. The system includes a first process (e.g., 216, page 11, lines 7-8) and a second processes (e.g., 210, page 11, lines 18-19), and memory (192, page 16, line 3), configured to provide asynchronous, interprocess communication between the first process and the second process, wherein the memory provides a portion of memory (240, 242, 244, page 12, lines 2-4) configured to be accessible by the first and second processes, wherein access to the portion of memory is granted exclusively to the first process (530, page 13, line 12) for modification of contents of the portion of memory (532, page 13, lines 13) to prevent inter-process communication between the first and second process from becoming a performance bottleneck by releasing resources of the first process

after the first process modifies the contents of the portion of memory (536, page 13, lines 14-15).

Independent claim 49 recites a service level agreement (SLA) server. The SLA server includes a plurality of processes (e.g., 216, page 11, lines 7-8 and 210, page 11, lines 18-19), the plurality of processes comprising a database manager (312, page 12, lines 6-7) for managing performance data, an application server (316, page 12, lines 7-9) for collecting performance data and providing a client interface for establishing service level agreements, a SLA core (314, page 12, lines 7) for analyzing data and controlling actions based on service level agreements and policy and a performance monitor daemon (340, page 12, lines 10) for communicating with remote I/O service gateways to collect data and send throttling requests and memory (192, page 16, line 3), configured to provide asynchronous, interprocess communication between the the processes, wherein the memory provides a portion of memory (240, 242, 244, page 12, lines 2-4) configured to be accessible by the processes, wherein access to the portion of memory is granted exclusively to a first process (530, page 13, line 12) for modification of contents of the portion of memory (532, page 13, lines 13) to prevent inter-process communication between the first and second process from becoming a performance bottleneck by releasing resources of the first process after the first process modifies the contents of the portion of memory (536, page 13, lines 14-15).

Independent claim 50 recites another embodiment of a service level agreement (SLA) server. The SLA server includes a processor (194, page 16, line 4) configured for providing a plurality of processes (e.g., 216, page 11, lines 7-8 and 210, page 11, lines 18-19) and memory (192, page 16, line 3), configured to provide asynchronous, interprocess communication between the the processes, wherein the memory provides a portion of

memory (240, 242, 244, page 12, lines 2-4) configured to be accessible by the processes, wherein access to the portion of memory is granted exclusively to a first process (530, page 13, line 12) for modification of contents of the portion of memory (532, page 13, lines 13) to prevent inter-process communication between the first and second process from becoming a performance bottleneck by releasing resources of the first process after the first process modifies the contents of the portion of memory (536, page 13, lines 14-15).

Independent claim 51 recites a method for providing a two-step communication scheme. The method includes establishing a portion of memory (240, 242, 244, page 12, lines 2-4) configured to provide asynchronous, connectionless inter-process communication between a first process (e.g., 216, page 11, lines 7-8) and a second processes (e.g., 210, page 11, lines 18-19), granting exclusive access (530, page 13, line 12) to a first process to the portion of memory configured to provide asynchronous, connectionless inter-process communication (240, 242, 244, page 12, lines 2-4) between the first process (e.g., 216, page 11, lines 7-8) and the second processes (e.g., 210, page 11, lines 18-19), while having been granted to the exclusive access to the portion of memory configured to provide asynchronous, connectionless inter-process communication, accessing the portion of memory (532, page 13, line 13) configured to provide asynchronous, connectionless inter-process communication by the first process to modify the contents thereof to provide a message for processing by the second process and releasing exclusive access (536, page 13, lines 14-15) by the first process to the portion of memory configured to provide asynchronous, connectionless inter-process communication to prevent inter-process communication between the first and second process from becoming a performance bottleneck by releasing

resources of the first process after the first process modifies the contents of the portion of memory.

Independent claim 52 recites another embodiment of a server. The server includes means for storing data (192, page 16, line 3), wherein a portion of the means for storing data (240, 242, 244, page 12, lines 2-4) is configured to provide two-step, asynchronous, connectionless inter-process communication between a first process (e.g., 216, page 11, lines 7-8) and a second processes (e.g., 210, page 11, lines 18-19), the portion of the means for storing data is configured to be accessible by the first and second processes, wherein access to the portion of the means for storing data is granted exclusively (530, page 13, line 12) to the first process for modification of contents of the portion of the means for storing data (532, page 13, lines 13) to prevent inter-process communication between the first and second process from becoming a performance bottleneck by releasing resources of the first process after the first process modifies the contents of the portion of the means for storing data (536, page 13, lines 14-15).

Independent claim 53 recites another embodiment of a system. The system includes first process means (e.g., 216, page 11, lines 7-8), second process means, (e.g., 210, page 11, lines 18-19), means for storing data (192, page 16, line 3), configured to provide asynchronous, interprocess communication between the first process means and the second process means, wherein the means for storing data is configured to be accessible by the first and second process means, wherein access to the portion of the means for storing data (240, 242, 244, page 12, lines 2-4) is granted exclusively (530, page 13, line 12) to the first process means for modification of contents of the portion of the means for storing data (532, page 13, lines 13) to prevent inter-process communication between the first and second process means

from becoming a performance bottleneck by releasing resources of the first process means after the first process means modifies the contents of the portion of the means for storing data (536, page 13, lines 14-15).

Independent claim 54 recites another embodiment of a service level agreement (SLA) server. The SLA server includes a plurality of process means (e.g., 216, page 11, lines 7-8 and 210, page 11, lines 18-19), the plurality of processes comprising process means for managing performance data (312, page 12, lines 6-7), process means for collecting performance data and providing a client interface for establishing service level agreements (316, page 12, lines 7-9), process means for analyzing data and controlling actions based on service level agreements and policy (314, page 12, lines 7) and process means for communicating with remote I/O service gateways to collect data and send throttling requests (340, page 12, lines 10) and means for storing data (192, page 16, line 3) configured to provide asynchronous, interprocess communication between the plurality of process mans, wherein the means for storing data is accessible by the plurality of process means (240, 242, 244, page 12, lines 2-4), wherein access to the means for storing data is granted exclusively to a first of the process means (530, page 13, line 12) for modification of contents of the means for storing data (532, page 13, lines 13) to prevent inter-process communication between the plurality of process means from becoming a performance bottleneck by releasing resources of the first of the process means after the first of the process means modifies the contents of the portion of means for storing data (536, page 13, lines 14-15).

VI. Grounds Of Rejections To Be Reviewed On Appeal

Appellant has attempted to comply with new rule 37 C.F.R. § 41.37(c) by providing the Office Action's grounds of rejection verbatim, followed by an argument section corresponding thereto.

A. Claims 31-48 and 51-53 were rejected under 35 U.S.C. § 102(e) as being anticipated by Vessey et al.

B. Claims 49, 50 and 54 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Tummalapalli in view of Vessey et al.

VII. Argument

A. REJECTION OF CLAIMS 31-48 AND 51-53 UNDER 35 U.S.C. 102(e) AS BEING ANTICIPATED BY VESSEY ET AL.

1. VESSEY ET AL.FAILS TO TEACH, DISCLOSE OR SUGGEST PROVIDING ASYNCHRONOUS, CONNECTIONLESS INTER-PROCESS COMMUNICATION BETWEEN A FIRST PROCESS AND A SECOND PROCESSES

Applicants' invention, for example as presented by claim 31, provides for asynchronous (non-blocking) interprocess communication between processes. A portion of memory is configured to provide asynchronous, connectionless inter-process communication between a first process and a second processes. Exclusive access is granted to a first process to the portion of memory. While exclusive access, the first process accesses the portion of memory to modify the contents thereof to provide a message for processing by the second process. Exclusive access by the first process to the portion of memory is released to prevent inter-process communication between the first and second process from becoming a performance bottleneck. The release of exclusive access of the portion of memory by the first process releases resources of the first process.

Accordingly, communication between processes is provided in a manner that is asynchronous and thus does not present a bottleneck.

Independent claims 37, 43 and 51-53 provide similar limitations.

In contrast, Vessey et al. disclose providing a connection between a first partition and a second partition using a memory region of a computer system. For example, in paragraph 0440, Vessey et al. teach that in each partition a user-mode dynamic link library (DLL) 3430, works in combination with a kernel-mode device driver 3440 to service requests made by one of the applications 3400a, 3400b to establish network communications (e.g., a socket connection) with the other application on the other partition.

Referring to paragraph 441, Vessey et al. teach that the shared memory service provider DLL 3430 and shared memory SPI client device driver 3440 work together to establish data structures in shared memory to emulate the requested network connection. By requiring a connection between the first partition and the second partition, a bottleneck could occur because the connection between the first partition and the second partition is maintained. Moreover, maintaining a connection ties up finite resources of applications that maintain a connection between them.

In addition, Vessey et al. does not disclose allowing exclusive access to shared memory by a partition. Rather, Vessey et al. discloses a lock mechanism. However, the lock mechanism merely limits modification of any given structure, page or table to only one partition. Nevertheless, dictating how only one partition owns and therefore may change the content of the shared memory is different from providing exclusive access to the shared memory. Vessey et al. merely provides read access to the message queue to a partition that does not own the message queue.

Still further, the lock mechanism disclosed by Vessey et al. teaches away from providing asynchronous interprocess communication. Vessey et al. provides a message queue area that is divided into n node output queues, each of which is dedicated to a different partition. However, a particular partition can only modify its associated node output queue.

Vessey et al. also fails to suggest releasing exclusive access by a first process to allow a second process to then obtain exclusive access to the same portion of memory and change the contents of the portion of memory, e.g., by providing results to the message left in the portion of memory by the first process. Again, Vessey et al. does not allow a partition that does not have ownership of a portion of the memory to alter the contents of that memory.

Accordingly, Applicants respectfully submit that independent claims 31, 37, 43 and 51-53 are patentable over Vessey et al.

Furthermore, dependent claims 32-36, 38-42 and 44-48 are also patentable over Vessey et al. because they incorporate all of the limitations of the corresponding independent claim 31, 37 and 43. Further dependent claims 32-36, 38-42 and 44-48 recite additional novel elements and limitations. Applicants reserve the right to argue independently the patentability of these additional novel aspects. Therefore, Applicants respectfully submit that dependent claims 32-36, 38-42 and 44-48 are patentable over the cited references, and request that the objections to the independent claims be withdrawn.

B. REJECTION OF CLAIMS 49, 50 AND 54 UNDER 35 U.S.C. § 103(a) AS BEING UNPATENTABLE OVER TUMMALAPALLI IN VIEW OF VESSEY ET AL.

Tummalapalli fails to to overcome the deficiencies of Vessey et al. Tummalapalli is merely cited as disclosing a database manager, a client interface for establishing SLAs and an

SLA cored for controlling thoughput according to the established SLAs. program storage device for providing access to a mailslot and a remote procedure call. However, Tummalapalli fails to disclose, teach or suggest providing a portion of memory configured to provide asynchronous, connectionless inter-process communication between a first process and a second processes. Tummalapalli also fails to disclose or suggest allowing exclusive access to shared memory by a partition. Tummalapalli also fails to suggest releasing exclusive access by a first process to allow a second process to then obtain exclusive access to the same portion of memory and change the contents of the portion of memory, e.g., by providing results to the message left in the portion of memory by the first process.

In fact, the final Office Action admits that Tummalapalli fails to suggest the memory arrangement recited in the independent claims. Consequently, Tummalapalli fails to suggest the elements reicted in claims 49, 50 and 54.

Accordingly, Applicants respectfully submit that independent claims 49, 50 and 54 are patentable over Vessey et al. and Tummalapalli

VIII. Claims Appendix

1-30. (Canceled)

31. (Previously Presented) A program storage device readable by a computer, the program storage device tangibly embodying one or more programs of instructions executable by the computer to perform a method for providing a two-step communication scheme, the method comprising:

establishing a portion of memory configured to provide asynchronous, connectionless inter-process communication between a first process and a second processes;

granting exclusive access to a first process to the portion of memory configured to provide asynchronous, connectionless inter-process communication between the first process and the second process;

while having been granted to the exclusive access to the portion of memory configured to provide asynchronous, connectionless inter-process communication, accessing the portion of memory configured to provide asynchronous, connectionless inter-process communication by the first process to modify the contents thereof to provide a message for processing by the second process; and

releasing exclusive access by the first process to the portion of memory configured to provide asynchronous, connectionless inter-process communication to prevent inter-process communication between the first and second process from becoming a performance bottleneck by releasing resources of the first process after the first process modifies the contents of the portion of memory.

32. (Previously Presented) The program storage device of claim 31 further comprising configuring the memory to provide header having an operation code and a parameter region interpreted according to the operation code.

33. (Previously Presented) The program storage device of claim 31, wherein the providing the message into the portion of memory by the first process further comprises initiating a remote procedure call.

34. (Previously Presented) The program storage device of claim 31 further comprising granting exclusive access to the second process to the portion of memory configured to provide asynchronous, connectionless inter-process communication, while having been granted to the exclusive access to the portion of memory, accessing the portion of memory by the second process to modify the contents thereof to access the message provided in the portion of memory by the first process and releasing exclusive access by the second process to the portion of memory.

35. (Previously Presented) The program storage device of claim 34 further comprising:

establishing exclusive access to the portion of memory by the second process;

accessing the portion of memory by the second process to provide a result message in response to the message placed in the portion of memory by the first process; and

providing by the second process a notification to the first process to check the portion of memory.

36. (Previously Presented) The program storage device of claim 31 further comprising providing by the first process a notification to the second process to check the portion of memory.

37. (Previously Presented) A server comprising a memory, wherein a portion of the memory is configured to provide two-step, asynchronous, connectionless inter-process communication between a first process and a second process, the portion of memory being configured as memory accessible by the first and second processes, wherein access to the portion of memory being granted exclusively to the first process for modification of contents of the portion of memory to prevent inter-process communication between the first and second process from becoming a performance bottleneck by releasing resources of the first process after the first process modifies the contents of the portion of memory.

38. (Previously Presented) The server of claim 37, wherein the portion of memory comprises a slot having a header comprising an operation code and a parameter region interpreted according to the operation code.

39. (Previously Presented) The server of claim 37, wherein the placing message into the portion of memory by the first process further comprises initiating a remote procedure call.

40. (Previously Presented) The server of claim 37, wherein the second process is granted exclusive access to the portion of memory configured to provide asynchronous, connectionless inter-process communication, accesses the portion of memory to modify the contents thereof to access the message provided in the portion of memory by the first process and releases exclusive access by the second process to the portion of memory.

41. (Previously Presented) The server of claim 40, wherein the second process is granted exclusive access to the portion of memory, accesses the portion of memory to provide a result message in response to the message placed in the portion of memory by the first process and provides a notification to the first process to check the portion of memory.

42. (Previously Presented) The server of claim 37, wherein the first process provides a notification to the second process to check the portion of memory.

43. (Previously Presented) A system, comprising:

a first process;

a second process; and

memory configured to provide asynchronous, interprocess communication between the first process and the second process, wherein the memory provides a portion of memory configured to be accessible by the first and second processes, wherein access to the portion of memory is granted exclusively to the first process for modification of contents of the portion of memory to prevent inter-process communication between the first and second process from becoming a performance bottleneck by releasing resources of the first process after the first process modifies the contents of the portion of memory.

44 (Previously Presented) The system of claim 43, wherein the portion of memory comprises a slot having a header comprising an operation code and a parameter region interpreted according to the operation code.

45. (Previously Presented) The system of claim 43, wherein the placing message into the portion of memory by the first process further comprises initiating a remote procedure call.

46. (Previously Presented) The system of claim 43, wherein the second process is granted exclusive access to the portion of memory configured to provide asynchronous, connectionless inter-process communication, accesses the portion of memory to modify the contents thereof to access the message provided in the portion of memory by the first process and releases exclusive access by the second process to the portion of memory.

47. (Previously Presented) The system of claim 46, wherein the second process is granted exclusive access to the portion of memory, accesses the portion of memory to provide a result message in response to the message placed in the portion of memory by the first process and provides a notification to the first process to check the portion of memory.

48. (Previously Presented) The system of claim 43, wherein the first process provides a notification to the second process to check the portion of memory.

49. (Previously Presented) A service level agreement (SLA) server, comprising:

a plurality of processes, the plurality of processes comprising a database manager for managing performance data, an application server for collecting performance data and providing a client interface for establishing service level agreements, a SLA core for analyzing data and controlling actions based on service level agreements and policy and a performance monitor daemon for communicating with remote I/O service gateways to collect data and send throttling requests; and

memory configured to provide asynchronous, interprocess communication between the processes, wherein the memory provides a portion of memory configured to be accessible by the processes, wherein access to the portion of memory is granted exclusively to a first of the processes for modification of contents of the portion of memory to prevent inter-process communication between the process from becoming a performance bottleneck by releasing resources of the first of the processes after the first of the processes modifies the contents of the portion of memory.

50. (Previously Presented) A service level agreement (SLA) server, comprising:

a processor configured for providing a plurality of processes; and

memory configured to provide asynchronous, interprocess communication between the first process and the second process, wherein the memory provides a portion of memory configured to be accessible by the first and second processes,

wherein the processor grants exclusive access to the portion of memory by the first process for modification of contents of the portion of memory to prevent inter-process communication between the first and second process from becoming a performance bottleneck by releasing resources of the first process after the first process modifies the contents of the portion of memory.

51. (Previously Presented) A method for providing a two-step communication scheme, comprising:

establishing a portion of memory configured to provide asynchronous, connectionless inter-process communication between a first process and a second processes;

granting exclusive access to a first process to the portion of memory configured to provide asynchronous, connectionless inter-process communication between the first process and the second process;

while having been granted to the exclusive access to the portion of memory configured to provide asynchronous, connectionless inter-process communication, accessing the portion of memory configured to provide asynchronous, connectionless inter-process communication by the first process to modify the contents thereof to provide a message for processing by the second process; and

releasing exclusive access by the first process to the portion of memory configured to provide asynchronous, connectionless inter-process communication to prevent inter-process communication between the first and second process from becoming a performance bottleneck by releasing resources of the first process after the first process modifies the contents of the portion of memory.

52. (Previously Presented) A server comprising a means for storing data, wherein a portion of the means for storing data is configured to provide two-step, asynchronous, connectionless inter-process communication between a first process and a second process, the portion of the means for storing data is configured to be accessible by the first and second processes, wherein access to the portion of the means for storing data is granted exclusively to the first process for modification of contents of the portion of the means for storing data to prevent inter-process communication between the first and second process from becoming a performance bottleneck by releasing resources of the first process after the first process modifies the contents of the portion of the means for storing data.

53. (Previously Presented) A system, comprising:

first process means;

second process means;

means for storing data configured to provide asynchronous, interprocess communication between the first process means and the second process means, wherein the means for storing data is configured to be accessible by the first and second process means, wherein access to the portion of the means for storing data is granted exclusively to the first process means for modification of contents of the portion of the means for storing data to prevent inter-process communication between the first and second process means from becoming a performance bottleneck by releasing resources of the first process means after the first process means modifies the contents of the portion of the means for storing data.

54. (Previously Presented) A service level agreement (SLA) server, comprising:

a plurality of process means, the plurality of processes comprising process means for managing performance data, process means for collecting performance data and providing a client interface for establishing service level agreements, process means for analyzing data and controlling actions based on service level agreements and policy and process means for communicating with remote I/O service gateways to collect data and send throttling requests; and

means for storing data configured to provide asynchronous, interprocess communication between the plurality of process mans, wherein the means for storing data is accessible by the plurality of process means, wherein access to the means for storing data is granted exclusively to a first of the process means for modification of contents of the means for storing data to prevent inter-process communication between the plurality of process means from becoming a performance bottleneck by releasing resources of the first of the process means after the first of the process means modifies the contents of the portion of means for storing data.

IX. Evidence Appendix

Appellant is unaware of any evidence submitted in this application pursuant to 37 C.F.R. §§ 1.130, 1.131, and 1.132.

X. Related Proceedings Appendix

As stated in Section II above, Appellant is unaware of any related appeals, interferences or judicial proceedings.